The system P-ND

This system is modeled after the similar system that I found in the fine book

Patrick Suppes,
"Introduction to Logic,"
D. van Nostrand Co.

The rules of inference given here are not a minimal set of rules: some of them can be derived from others. This is a rather baroque, but convenient, set of rules.

In the rule-schemas below n, m, o, p are line numbers; A, B, C are propositions; d, d1, d2 are sets of line numbers. A --- means we don’t care about the contents of this slot.

A line in a derivation has four entries

A line number for this line.

A proposition.

An argument showing how this line was derived from previous lines.

A set of line numbers of the premises supporting this line.

Sentential Rules

0. Introduction of a premise

This rule requires no antecedents.

```
----------------------------------------
n     A              Premise     {n}
```

1. Modus Ponens

```
n     A-->B           ---         d1
m     A               ---         d2
----------------------------------------
o     B               (MP n m)    d1Ud2
```

2. Modus Tollens

```
n     A-->B           ---         d1
m     -B              ---         d2
----------------------------------------
o     -A              (MT n m)    d1Ud2
```

3. Conditional Proof

```
n     A               Premise     {n}
m     B               ---         d
----------------------------------------
o     A-->B           (CP m n)    d - {n}
```

4. Simplification

```
n     A&B              ---        d
--------------------------------------
m     A                (-&:L n)   d


n     A&B              ---        d
--------------------------------------
m     B                (-&:R n)   d
```

5. Adjunction

```
n     A                ---        d1
m     B                ---        d2
--------------------------------------
o     A&B              (+& n m)   d1Ud2
```

6. Addition

```
n     A                 ---        d
--------------------------------------
m     AvB               (+v:L n)   d


n     B                 ---        d
--------------------------------------
m     AvB               (+v:R n)   d
```

7. Modus Tollendo Ponens

```
n     AvB            ---        d1
m     -A             ---        d2
--------------------------------------
o     B          (MTP:L n m)    d1Ud2
```

```
n     AvB            ---        d1
m     -B             ---        d2
--------------------------------------
o     A          (MTP:R n m)    d1Ud2
```

8. Double Negation Elimination

```
n     --A             ---        d
--------------------------------------
m     A             (DNE n)      d
```

9. Double Negation Introduction

```
n     A               ---        d
--------------------------------------
m     --A           (DNI n)      d
```

10. Reducto ad absurdum

```
n     A          Premise      {n}
m     B          ---          d1
o     -B         ---          d2
---------------------------------------
p     -A      (RAA n m o)     d1Ud2-{n}
```

11. Biconditional-conditional

```
n     A<-->B          ---        d
-------------------------------------
m     A-->B           (-IFF:L n) d


n     A<-->B          ---        d
-------------------------------------
m     B-->A           (-IFF:R n) d
```

12. Conditional-biconditional

```
n     A-->B          ---        d1
m     B-->A          ---        d2
-------------------------------------
o     A<-->B     (+IFF n m)   d1Ud2
```

13. De Morgan’s laws

```
n     -(AvB)          ---        d
--------------------------------------
m     -A&-B           (Nv&N n)   d


n     -A&-B           ---        d
--------------------------------------
m     -(AvB)          (&NNv n)   d


n     -(A&B)          ---        d
--------------------------------------
m     -Av-B           (N&vN n)   d


n     -Av-B           ---        d
--------------------------------------
m     -(A&B)          (vNN& n)   d
```

14. Separation of cases

```
n      A-->B         ---        d1
m     -A-->B         ---        d2
--------------------------------------
o      B         (SCA n m)   d1Ud2


n      A-->C         ---        d1
m      B-->C         ---        d2
--------------------------------------
o     (AvB)-->C   (SCB n m)    d1Ud2
```

Rules for Quantifiers

In the following rules S[t;x;P] is the expression resulting from substituting t for each occurrence of x in P. Our rules of formation prevent the creation of an expression in which the same identifier is bound more than once. This prevents collision of variables, without restricting the expressive power of the language, because the meaning of any expression is independent of the names of the bound identifiers. The rules have auxilliary restrictions, which are stated in each case. In this formulation we introduce a syntactic device called an "arbitrary individual," which is syntactically distinguished by a prefix of *. In other formulations these are not distinguished from variables, leading to considerably hairier and less intuitively clear restrictions on the applicability of the rules.

15. Universal Specification

Restriction: In this rule t may be a term composed of constants or arbitrary individuals.

```
n     All x P        ---          d
--------------------------------------
m     S[t;x;P]    (US n t x)      d
```

16. Universal Generalization

Restriction: Here t must be some arbitrary individual not occurring in any line of d, and x may not appear in P.

```
n     P                ---        d
--------------------------------------
m    All x S[x;t;P]   (UG n x t)  d
```

17. Existential Specification

Restriction: In this rule t is a term composed of a new function symbol applied to the list of all of the arbitrary individuals occurring in P.

```
n     Exists x P     ---          d
-----------------------------------------
m     S[t;x;P]    (ES n t x)      d
```

18. Existential Generalization

Restriction: Here t is a term which contains no variables, and x may not appear in P.

```
n     P                 ---        d
-----------------------------------------
m    Exists x S[x;t;P] (EG n x t)  d
```

19. Quantifier Interchange

```
n    All x All y P     ---         d
--------------------------------------
m    All y All x P     (AA n)      d
```

```
n  Exists x Exists y P    ---     d
--------------------------------------
m  Exists y Exists x P    (EE n)  d
```

19. Quantifier Negation

```
n    -Exists x -P      ---         d
--------------------------------------
m    All x P           (-:-E- n)  d
```

```
n    -All x -P         ---         d
--------------------------------------
m    Exists x P        (-:-A- n)  d
```

```
n    Exists x P        ---         d
--------------------------------------
m    -All x -P         (+:-A- n)  d
```

```
n    All x P           ---         d
--------------------------------------
m    -Exists x -P      (+:-E- n)  d
```